NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN ENHANCED IMPLEMENTATION OF MODELS FOR ELECTRIC POWER GRID INTERDICTION

by

David D. Carnal

September 2005

Thesis Advisor:	R. Kevin Wood
Second Reader:	Javier Salmeron

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE September 2005	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE: Title (Mix case letters) An Enhanced Implementation Of Models For Electric Power Grid Interdiction		5. FUNDING NUMBERS	
6. AUTHOR(S) David D. Carnal			

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
   Naval Postgraduate School
   Monterey, CA 93943-5000
		8. PERFORMING ORGANIZATION REPORT NUMBER	

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE	

13. ABSTRACT (maximum 200 words)

   This thesis evaluates the ability of the Xpress-MP software package to solve complex, iterative mathematical-programming problems. The impetus is the need to improve solution times for the VEGA software package, which identifies vulnerabilities to terrorist attacks in electric power grids. VEGA employs an iterative, optimizing heuristic, which may need to solve hundreds of related linear programs. This heuristic has been implemented in GAMS (General Algebraic Modeling System), whose inefficiencies in data handling and model generation mean that a modest, 50-iteration solution of a real-world problem can require over five hours to run. This slowness defeats VEGA's ultimate purpose, evaluating vulnerability-reducing structural improvements to a power grid.

   We demonstrate that Xpress-MP can reduce run times by 60%-85% because of its more efficient data handling, faster model generation, and the ability, lacking entirely in GAMS, to solve related models without regenerating each from scratch. Xpress-MP's modeling language, Mosel, encompasses a full-featured procedural language, also lacking in GAMS. This language enables a simpler, more modular and more maintainable implementation.

   We also demonstrate the value of VEGA's optimizing heuristic by comparing it to rule-based heuristics rules adapted from the literature. The optimizing heuristic is much more powerful.

14. SUBJECT TERMS Electric Power Girds; Network Interdiction; Mixed Integer Programming; GAMS; Xpress-MP			15. NUMBER OF PAGES 61
			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

THIS PAGE INTENTIONALLY LEFT BLANK

AN ENHANCED IMPLEMENTATION OF MODELS FOR ELECTRIC POWER GRID INTERDICTION

David D. Carnal
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1996

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 2005

Author: David D. Carnal

Approved by: R. Kevin Wood
               Thesis Advisor

               Javier Salmeron
               Second Reader

               James Eagle
               Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis evaluates the ability of the Xpress-MP software package to solve complex, iterative mathematical-programming problems. The impetus is the need to improve solution times for the VEGA software package, which identifies vulnerabilities to terrorist attacks in electric power grids. VEGA employs an iterative, optimizing heuristic, which may need to solve hundreds of related linear programs. This heuristic has been implemented in GAMS (General Algebraic Modeling System), whose inefficiencies in data handling and model generation mean that a modest, 50-iteration solution of a real-world problem can require over five hours to run. This slowness defeats VEGA's ultimate purpose, evaluating vulnerability-reducing structural improvements to a power grid.

We demonstrate that Xpress-MP can reduce run times by 60%-85% because of its more efficient data handling, faster model generation, and the ability, lacking entirely in GAMS, to solve related models without regenerating each from scratch. Xpress-MP's modeling language, Mosel, encompasses a full-featured procedural language, also lacking in GAMS. This language enables a simpler, more modular and more maintainable implementation.

We also demonstrate the value of VEGA's optimizing heuristic by comparing it to rule-based heuristics rules adapted from the literature. The optimizing heuristic is much more powerful.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to acknowledge the faculty at the Naval Postgraduate School who provided me with the knowledge to undertake this work. Specifically, Professor Kevin Wood whose professionalism and analytic insight ensured the highest quality end product was achieved. His tireless dedication and patience made it all possible. You have been an outstanding advisor and I could not have done it without you.

Secondly, I would like to thank Professor Javier Salmeron who introduced me to this topic and played a major role in my decision to work on this project. He provided the ground work and direction to ensure this thesis would be great. His support was instrumental in the successful completion of this thesis.

Finally, I would like to thank my family that supported me through this challenging time, specially my wife Holly who provided encouragement and held our home together while I undertook this study. Thank you all.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

This thesis enhances an existing software package for analyzing the vulnerability of electric power grids to terrorist attacks.

Since the terrorist attacks on 11 September 2001, the U.S. is reassessing the vulnerability of its critical infrastructure. Electric power grids are key infrastructure systems that are critical to the United States' economy and security.

The VEGA tool, developed by researchers at the Naval Post Graduate School and the University of Texas at Austin, is an integrated system for analyzing the vulnerability of electric power-transmission grids to terrorist attacks. At VEGA's core is an optimization model that posits terrorists with limited offensive resources to carry out physical attacks on a given power grid. Solution of the model identifies components in order to maximize disruption, i.e., unserved demand for energy. We deem substations, transformers, buses, lines, and generators to be interdictable components, provided sufficient interdiction resource is applied.

This thesis seeks to improve computation times for the Interdicting DC Optimal Power-Flow Heuristic (IDCH), which approximately solves the optimization model within VEGA. IDCH is currently implemented in the General Algebraic Modeling System (GAMS) and uses a highly efficient solver, CPLEX. However, due to inefficient model generation and data handling, GAMS can take hours to analyze a grid and identify critical components. We implement IDCH in Xpress-MP, a powerful optimization software package that is more efficient in generating models and handling data. Average run times are reduced by 75.7% with Xpress-MP; the greatest reduction is 84.6%. The thesis also points out some of the other advantages of Xpress-MP, which include dynamic arrays, functions and procedures, and compilation.

This thesis also investigates the actual effectiveness of IDCH for analyzing the vulnerability of an electrical power grid to terrorist attacks. To do this, we compare the interdiction plans generated by IDCH to the plans produced by adapting the heuristic rules proposed by Albert, Albert and Nakarado (AAN) in their work "Structural

Vulnerability of the North American Power Grid." AAN ignore electrical-engineering realities in their analysis and use heuristic rules in an attempt to find interdiction plans that maximize a simple surrogate for disruption (i.e., unmet demand for energy). We implement variations of their heuristic rules that are sensible from an electrical engineering standpoint, and measure effectiveness in terms of realistic estimates of disruption, such as unserved energy. Overall, the AAN rules perform poorly and inconsistently compared to IDCH.

I. INTRODUCTION

The VEGA software package (Vulnerability of Electric Power Grids Analyzer), developed at the Naval Postgraduate School and the University of Texas at Austin [Salmeron et al. 2004A, 2005], is an integrated system for analyzing the vulnerability of an electric power-transmission grid to a coordinated terrorist attack. At VEGA's core is an optimization model whose solution identifies a set of grid components that is most critical to system functionality: For fixed levels of attack resources, one set of components is more critical that another if (a) either set can be feasibly attacked and disabled, and (b) disabling the first set causes more "disruption" than disabling the latter. Disruption is measured in terms of unserved demand for energy. This thesis seeks to (a) improve computation times for the heuristic algorithm currently used to approximate the solution to the optimization model, and (b) demonstrate the value of these improvements when analyzing real-world transmission grids.

A. BACKGROUND

Since the terrorist attacks on 11 September 2001, the U.S. is reassessing the vulnerability of its critical infrastructure. Electric power grids are key infrastructure systems that are vulnerable to terrorist attacks, and the Department of Homeland Security has taken note [Department of Homeland Security 2003].

In the past, an electric-power utility company was concerned with the vulnerability of its system to natural disasters, unplanned outages caused by equipment failures, and minor man-made problems (e.g., cars running into power poles, people with rifles taking pot shots at insulators). Today, a utility must also be concerned with the prospect of multiple, simultaneous attacks on important equipment in its system. This threat is real given that (a) U.S. forces in Afghanistan discovered Al Qaeda documentation about a facility that controls power distribution for the eastern U.S., and (b) maps of the U.S. electrical transmission grid are publicly available for less than $100 on the Internet [Energy Pulse 2003].

1

Vulnerability to attack has increased in recent years, too. As demand for electricity rises, the reserve levels in transmission capacity decrease unless adequate capacity is added. But adequate transmission capacity has not been added [Report of the National Energy Policy Development Group 2001]. This means that the "safety cushions" that utilities have built into their systems to handle failures, intentionally caused or not, have diminished. Clearly, utilities have an increased need to be able to analyze the adequacy of their reserve levels with respect to potential failures.

The discussion above motivates the development of optimization models to represent the problem of terrorists attacking a power system. By studying how to attack power grids, insight can be gained about how to protect them.

There are many types of "vulnerability analyses" in the electric power industry (see the overview in NERC [2002]), but only VEGA quantifies the amount of unserved demand that would accrue from a worst-case attack (Salmeron et al. [2004A, 2004B]). VEGA quantifies this through a bilevel optimization model, although only heuristic solution procedures have proven viable for realistic problems. VEGA's Interdicting Optimal DC Power-Flow Heuristic (IDCH) repeatedly solves two submodels: a linear program (LP) known as DC Optimal Power-Flow (DCOPF), and a mixed integer program (MIP) known as the "interdiction approximating master problem" (IAMP).

All submodels solved in IDCH are currently generated using the General Algebraic Modeling System (GAMS) [2004] and solved with CPLEX [2004] or any other MIP solver which can be called from GAMS. This thesis looks to reduce computation times for this procedure by implementing it entirely using the Xpress-MP optimization software [Dash 2005]. All submodels are written and generated using Xpress's Mosel algebraic modeling language, and are solved using the Xpress-MP solver. Reducing computation time is crucial because it allows VEGA's users to:

(a) solve more problem instances (e.g., under different assumptions on load conditions, terrorist resources, etc.), and

(b) run more iterations, which may improve the accuracy of solutions.

In turn, (a) and (b) allow us to better assess which components are in greater need of protection, which is VEGA's ultimate purpose.

The current GAMS implementation spends approximately 90% of its total computation time handling data (i.e., reading the data, creating intermediate data structures, reading the solution, and creating the solution output) and model generation (i.e., creating the structures required for the problem to be solved by the chosen solver). For example, every iteration of the algorithm, when applied to a model of the ERCOT grid (Electric Reliability Council of Texas), requires one minute of CPU time, of which only 4 seconds are devoted to actually solving the model.

The Mosel technology within Xpress-MP is similar to structured programming languages in that it allows writing and using functions and procedures, while being able to embed mathematical programs. This allows IDCH to be modularized, which will make future improvements easier to incorporate. GAMS supports neither functions nor procedures.

GAMS does use concise algebraic statements to define models, but the language limits the user's ability to control data structures. Mosel allows a great deal of control over data structures while also using concise algebraic statements to define models. As one example, Mosel efficiently implements an arbitrary index calculation into a multidimensional array.

A second part of this thesis demonstrates IDCH's capabilities. We compare results that IDCH achieves to the results obtained using variants of the methods described by Albert, Albert, and Nakarado [2004] in their paper "Structural Vulnerability of the North American Power Grid." For simplicity, we shall often refer to this paper and its authors as "AAN."

AAN measure network functionality using crude connectivity measures in the grid rather than measuring how well the grid actually performs (e.g., the fraction of demanded energy that is actually supplied). Furthermore, AAN make no attempt to determine worst-case attacks in terms of their surrogate for grid functionality: They only show that one heuristic rule seems to be better than two other rules, when measured using their surrogate. This thesis will (a) simulate the experiments performed by AAN—for

example, measure changing system functionality as substations are "interdicted," i.e., attacked and disabled, following a specific rule—but use justifiable, electrical-engineering concepts to measure system functionality, and (b) compare those results to IDCH's results.

AAN perform their analysis on the entire North American power grid, which consists of over 10,000 generating units having a total production in excess of 760 gigawatts (GW), and over 40,000 transmission lines. Conducting this type of analysis on such a large grid would be a daunting task for us, but it would also be misleading. The grid is actually divided into three main sub-systems: the Eastern interconnection, the Western interconnection and the Texas interconnection. These systems have only modest interconnection capabilities, i.e., they operate nearly independently, and thus are never analyzed together by power engineers [North American Energy Working Group 2002]. Therefore, this thesis will only analyze individual "interconnections," specifically the Western and Texas interconnections.

B.   THESIS OUTLINE

Subsequent chapters in this thesis are organized as follows: Chapter II introduces the DCOPF, which is the basis for the interdiction model used in VEGA. That chapter also defines IAMP and describes the heuristic solution procedure currently in use. Chapter III then shows how Xpress-MP makes IDCH run more efficiently, and provides a detailed comparison of solution times between GAMS and Xpress-MP. Chapter IV provides additional computational results, including the comparisons with the techniques suggested by AAN [2004]. Chapter V provides a summary of results.

# II.	A HEURISTIC FOR OPTIMIZING INTERDICTIONS OF ELECTRICAL POWER GRIDS

This chapter introduces the Interdicting DC Optimal Power-Flow Heuristic developed by Salmeron et al. [2004A, 2004B, 2005], although first named "IDCH" here. IDCH incorporates a DCOPF model and an IAMP. DCOPF is a LP that is solved many times using data that represent different states of the power grid after a set of components is disabled by interdiction; and a set of such LPs must be solved for each of a set of interdiction plans. IAMP is a MIP whose solution identifies a set of components whose interdiction is (a) resource-feasible, (b) consistent with respect to certain logical restrictions, (c) never repeats a previously generated interdiction plan, and (d) maximizes the sum of estimated "component-interdiction values."

## A.	BACKGROUND

Salmeron et al. [2004B, 2005] describe an exact interdiction model that represents an instance of a bilevel mixed-integer program (e.g., Bard and Moore [1990]). In theory, this model identifies the maximum amount of disruption that a group of terrorists could cause to a power grid using limited interdiction (offensive) resources. However, the model is currently too difficult to solve exactly, by direct means or through decomposition, using data for a realistically sized electrical grid. As a result, this thesis focuses on applying the heuristic, IDCH. Although solutions found by IDCH may not be optimal, Salmeron et al. [2004A, 2005] show that IDCH provides good results for two small IEEE Reliability Test Systems. Thus, we expect it to perform well on larger, real-world grids.

The IDCH algorithm is straightforward, and may be viewed in terms of two competing "players." On one side is a power-system operator (SO) who wants to meet all regional demand for energy, while minimizing production costs. On the other side is a group of terrorists that wants to apply its limited resources to cause maximal damage to the grid, i.e., maximize the amount of unserved demand for energy, or its cost to society. Both sides are assumed to have perfect information as to how power can be transferred through the grid.

IDCH begins with the SO finding the cheapest way to produce enough energy to meet demand over a given period of time. For simplicity, let us assume that demand and costs are constant over that period, so that the SO can optimize the operation by solving one instance of DCOPF. (DCOPF minimizes the instantaneous cost of power generation, but since demand and costs are constant, results differ only by the multiplicative factor of "hours.") Typically, no unserved demand arises at this point, but if that becomes unavoidable, DCOPF will minimize the cost of generation plus the penalty for unserved demand.

The terrorists then replicate the SO's solution, and examine the power flows in the grid. From these values, they estimate the importance of each component, to system functionality, on an individual basis; for instance, this estimated value might simply be the load (which is equivalent to energy) carried by the component, be it a transformer, substation, bus, line or generator. (Strictly speaking a substation is a collection of components, but can be viewed as a single component for purposes of interdiction.) The terrorists then determine "the estimated most-valuable" set of components that they can feasibly interdict, i.e., without exceeding their interdiction assets. The interdiction is carried out, and the SO responds optimally to the loss of the interdicted components. He does this by solving a new instance of DCOPF, one that reflects the newly interdicted components, and by then implementing the model-suggested generation levels, which, in turn, lead to the flows predicted by DCOPF. Then, the terrorists view the new power flows, make new estimates of component values, and select a new feasible interdiction plan without duplicating any of their previous plans. This process repeats for a fixed number of iterations, and the interdiction plan that yields the most unmet demand for energy is deemed an approximation to the optimal solution of the interdiction problem.

B.   POWER-FLOW MODEL

The DCOPF model forms the backbone of IDCH. DCOPF is a linear program that minimizes instantaneous generation cost plus the penalty associated with unmet load. The model provides an approximation of an exact nonlinear model, but the approximation is adequate for high-level security analyses such as ours [Wood and Wollenberg 1996, p. 514]. We represent DCOPF as a standard LP:

$$\text{DCOPF:} \qquad f = \min_x \mathbf{cx}$$

$$\text{s.t. } A\mathbf{x} = \mathbf{b}$$

$$\mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$$

where A is an $m \times n$ matrix, and all vectors conform. The constraints $A\mathbf{x} = \mathbf{b}$ represent flow balance constraints for power, and admittance constraints. The variables $\mathbf{x}$ represent power generation and flows, unmet demands, and phase angles. In the actual model, some variables will have $-u_j \leq x_j \leq u_j$, but any LP can be converted to one in which all variables are non-negative as shown. Appendix A contains a detailed formulation for this model as used in Salmeron et al. [2004A, 2004B, 2005].

The demand on an electrical grid varies throughout the day as can generation costs and unmet demand penalties. Therefore, measuring the cost of supplying and not supplying energy (i.e., power integrated over time), over a 24-hour period, provides a better measure of the vulnerability of the grid. This modification of the basic model is handled through a standard load-duration curve (LDC). The LDC approximates continuously varying data by (a) positing a set of time periods $p = 1,\ldots,P$, with durations t_p, such that $\sum_p t_p = 24$ hours, (b) defining a constant cost and penalty vector $\mathbf{c}_p$ for each period p, (c) defining a constant demand vector $\mathbf{d}_p$ for each p, and (d) incorporating $\mathbf{d}_p$ into a right-hand-side vector $\mathbf{b}_p$. (A simplistic approximation of a LDC for a one-day period could consist of $P = 3$ "segments," representing "peak," "standard," and "valley" loads.) If f_p now denotes the cost of supplying energy, the multi-period version of DCOPF is:

$$\text{For } p = 1,\ldots,P, \qquad f_p = t_p \min_{\mathbf{x}} \mathbf{c}_p \mathbf{x}_p$$

$$\text{s.t. } A\mathbf{x}_p = \mathbf{b}_p$$

$$\mathbf{0} \leq \mathbf{x}_p \leq \mathbf{u}.$$

In the presence of interdiction, the multi-period DCOPF must be extended over multiple days to account for differences in component repair times (e.g., a line might require 48 hours to repair, a bus might require 168 hours). Thus, the index p ("time period") now represents the length of time the grid is in a particular state of repair and subject to a particular segment of the LDC. "Period" could also cover within-week

7

variations, as well as seasonal demand variations if component repair times were to extend for months.

An interdiction plan $\boldsymbol{\delta}$ is a binary vector defined such that $\delta_k = 1$ if component k is interdicted, and $\delta_k = 0$ otherwise. A component that is interdicted forces certain variables and constraints to be eliminated. For instance, if a substation is interdicted, all lines connected to that substation, along with associated flow variables and/or admittance constraints, must be eliminated from the problem. One way to represent this is:

$$f_p\left(I_p, \boldsymbol{\delta}\right) = t_p \min_{\mathbf{x}} \mathbf{c}_p \mathbf{x}_p$$
$$\text{s.t. } A_i \mathbf{x}_p = \left(\mathbf{b}_p\right)_i \ \forall i \notin I_p\left(\boldsymbol{\delta}\right) \qquad (2.1)$$
$$\mathbf{0} \leq \mathbf{x}_p \leq \mathbf{u}_p\left(\boldsymbol{\delta}\right),$$

where

- $A_i \mathbf{x}_p = \left(\mathbf{b}_p\right)_i$ is the i-th row of $A\mathbf{x}_p = \mathbf{b}_p$,

- $I_p\left(\boldsymbol{\delta}\right)$ is the index set for constraints that must be eliminated in period p if interdiction plan $\boldsymbol{\delta}$ is carried out, and

- $\left(\mathbf{u}_p\left(\boldsymbol{\delta}\right)\right)_j = 0$ if x_j must be eliminated given $\boldsymbol{\delta}$, and $\left(\mathbf{u}_p\left(\boldsymbol{\delta}\right)\right)_j = u_j$ otherwise.

Note that the above representation excludes unnecessary constraints induced by $i \in I_p\left(\boldsymbol{\delta}\right)$ but, for simplicity, all original variables x_j are maintained in the formulation, with "eliminated" variables being fixed to zero.

The total cost of an interdiction plan interdiction $\boldsymbol{\delta}$ is

$$F\left(\boldsymbol{\delta}\right) = \sum_p f_p\left(I_p, \boldsymbol{\delta}\right). \qquad (2.2)$$

And, the interdiction problem we (actually, the terrorists) would like to solve is

$$\text{I-DCOPF: } \max_{\boldsymbol{\delta} \in \Delta} F\left(\boldsymbol{\delta}\right), \qquad (2.3)$$

where $\boldsymbol{\delta} \in \Delta$ represents interdiction-resource constraints plus the fact that interdictions are binary decisions.

C.   AN OPTIMIZING INTERDICTION HEURISTIC (IDCH)

The heuristic outlined in Salmeron et al. [2004A, 2004B, 2005] to solve I-DCOPF approximately has two parts that are solved repeatedly: (a) IAMP finds a resource-feasible interdiction plan $\boldsymbol{\delta}$ that maximizes the sum of estimated values for individual interdictions, subject to some logical constraints, and (b) $F(\boldsymbol{\delta})$ is evaluated by solving multiple instances of DCOPF, once for each period corresponding to a repair state of the grid and a segment of the LDC. The heuristic, IDCH is summarized below. Note that "the full study length" denotes the maximum length of time that might be required to repair all interdicted components.

Outline of IDCH

(a)   Set iteration $\tau := 1$, $\boldsymbol{\delta}^* := \hat{\boldsymbol{\delta}}^\tau := \mathbf{0}$ and evaluate $F(\hat{\boldsymbol{\delta}}^\tau)$, i.e., evaluate the cost of operating the grid (including any unmet demand penalty, although unmet demand is unlikely given $\hat{\boldsymbol{\delta}}^\tau := \mathbf{0}$) over the full study length given no interdictions. Set $F^* := F(\hat{\boldsymbol{\delta}}^\tau)$.

(b)   Use the power-flow patterns at the current iteration to calculate the "value" V_k of each component k. V_k depends on the power flow through, out of, or into a component and reflects repair time. (Components that require a long time to repair are intrinsically more valuable to the terrorists.) The values are actually computed as moving averages, except that if a component is interdicted in iteration τ, its value remains the same in iteration $\tau + 1$.

(c)   Set $\tau := \tau + 1$ and solve IAMP for $\hat{\boldsymbol{\delta}}^\tau$:

$$\max_{\boldsymbol{\delta}} \sum_k V_k \delta_k$$

s.t.   Interdiction resources are not exceeded,

Components that are indirectly interdicted are not directly interdicted,

No previous interdiction plan, $\hat{\boldsymbol{\delta}}^2, \ldots, \hat{\boldsymbol{\delta}}^{\tau-1}$, is repeated, and

All variables δ_k are binary, 0-1.

9

(d)   Evaluate $F(\hat{\boldsymbol{\delta}}^{\tau})$.   If $F(\hat{\boldsymbol{\delta}}^{\tau}) > F^*$, then $\hat{\boldsymbol{\delta}}^{\tau}$ is the best interdiction plan found thus far, so set $F^* := F(\hat{\boldsymbol{\delta}}^{\tau})$ and $\boldsymbol{\delta}^* := \hat{\boldsymbol{\delta}}^{\tau}$.

(e)   If stopping criteria are satisfied stop, else return to step (b).

The detailed mathematical formulation of the IAMP is presented in Appendix B.

III. IMPLEMENTATION OF INTERDICTING OPTIMAL POWER FLOW HEURISTIC IN XPRESS-MP

Section A of this chapter introduces the Xpress-MP/Mosel software package ("Xpress") as an alternative to GAMS, and describes differences between these packages. Section B describes the implementation of the DCOPF subproblem in Xpress, and Section C describes the implementation of IDCH's master problem, IAMP. Section D provides a comparison of computing times for IDCH using Xpress and GAMS. Section E provides some programming tips regarding Mosel.

A. BACKGROUND ON MOSEL AND XPRESS-MP

Mosel is the algebraic modeling language included in the Xpress "package," designed and sold by Dash Optimization [Xpress-Mosel Users Guide 2004]. Mosel allows a mathematical program to be written in concise algebraic statements which generate one or more model instances. Mosel calls the Xpress-MP solver to solve those instances. This thesis uses Xpress as an alternative to GAMS because Xpress incorporates dynamic objects, functions and procedures, efficient model generation, and compilation, which GAMS does not incorporate.

1. Dynamic Objects

Mosel supports the use of dynamic objects which do not require pre-declaration of an object's size. This allows data structures to be smaller in Mosel than the corresponding structures in GAMS. Because Mosel has more efficient data structures, it can execute loops more efficiently than GAMS. Such loops are used repeatedly when generating a single model instance, let alone many model instances.

2. Functions and Procedures

Mosel supports the use of functions and procedures. Functions and procedures are subroutines that can be called, with arguments, by the main part of a program or by another subroutine; a function returns a value while a procedure does not. The use of functions and procedures allows for this program to be built in modules, which simplifies maintenance and updates. GAMS supports neither functions nor procedures.

3. Model Generation

Every time GAMS solves a model instance, it must "generate" the instance. That is, it must create, from scratch, the data structures, as files, that define the model for the solver. If modeling is carried out appropriately, Mosel does not need to regenerate a model that is a slight modification of a previously generated-and-solved model. Modifications are made by changing bounds on decision variables and by changing parameters associated with decision variables in the objective function.

Xpress has another advantage in that its generation engine, Mosel, and its solver, Xpress-MP, are tightly coupled. As a result, the interfacing between the generator and solver is carried out without the need for intermediary files whose use entails inefficiencies.

4. Compilation

Mosel is a programming language, and models written in Mosel are compiled prior to execution. When a Mosel (.mos) file is compiled it creates a Binary Model (.bim) file. The .bim files contains a (semi-)compiled version of the .mos file. In this form, the model is ready to be executed and the .mos file is not required anymore. To actually "run" the model, the .bim file must be read in again by Mosel and then executed [Xpress-Mosel Language Reference Manual 2003]. This allows for the author(s) of the model to protect their intellectual property, while distributing their model to various users without providing the source code.

GAMS also has this capability (i.e., secure work files), if the user has the proper "privacy license." However, GAMS requires additional work to produce the secure work files because the privacy license corresponds to the GAMS license of the individual user who is going to run the program [GAMS 2001]. This means that one program file must be created and distributed to each user. Furthermore, if the user upgrades GAMS and the license changes, the work file must be rebuilt and redistributed to that user. In contrast, the .bim file from Xpress can be used by anyone that has a current Xpress license.

5. Other Advantageous Features of Mosel, and One Disadvantage

Mosel easily reads input files. The following example shows Mosel's format for input of data from a file and an example of the code that reads the data from the file.

```
Bus_par: [
('1')        [false       true        2       168]
('1002')     [false       true        2       168]
('1003')     [false       true        2       168]
('1004')     [false       true        2       168]
...]


initializations from 'data\Bus_par.dat'
     [Angle_fixed, B_interdictable, Int_res_bus, Int_dur_bus] as
     'Bus_par';
end-initializations
finalize(Bus);
```

In this example, four different parameters for buses are imported into the respective declared arrays for holding the data. As the data is read from the file, the elements of the set `Bus` (e.g., 1, 1002, 1003, 1004) are defined. The `finalize()` command prevents new elements from being added to the set. (This command is optional, but ensures that a new element is not added to the set inadvertently; furthermore, operations with a "finalized" set can be carried out more efficiently.) This useful feature in Mosel means that the elements of a set need not be defined in advance of the data that refers to the set. GAMS requires that all elements of a set be defined prior to using that set.

GAMS also reads input files, but it must be done carefully because GAMS requires the use of the `include` command. This command operates like the "paste" function of any word processor. It literally "pastes" the file's data into the source code at the `include` statement's location. Mosel also has an `include` command that operates in this manner, but it has an input file format and command that reads the file, as shown in the example above, that is much more flexible. This flexibility is shown by IDCH only requiring 21 data files for the Xpress version as opposed to the 29 files required in the GAMS version.

Another useful feature of Xpress-MP is that a model is easily called by a Java, VB, or C++ program. This thesis utilizes this feature to run the IDCH multiple times with differing levels of interdiction resource. We also use this feature to integrate the Xpress programs into VEGA as an embedded function, rather than as external programs, as necessitated by GAMS.

In our experience, the only disadvantage of modeling with Mosel, is that it forces the user to use the Xpress-MP solver. GAMS allows the user to select from a variety of solvers, e.g. CPLEX, XA, and even Xpress-MP.

B.   IMPLEMENTATION OF THE DCOPF SUBPROBLEM IN IDCH

The mathematical formulation of DCOPF is covered in the previous chapter and in Appendix A. For each iteration τ of IDCH, one instance of DCOPF must be solved for each repair-state of the network and for each segment of the LDC. In our test problem, the LDC has three segments, and up to three repair states. Thus, up to nine instances of DCOPF must be solved in each iteration of IDCH.

Our computational experience shows that it is most efficient to generate, from scratch, a "baseline instance" of DCOPF in each iteration of IDCH, but only one. Thus, each iteration generates (and solves) an instance for some period $p = 1$, but does not regenerate the model instances for $p = 2,\ldots,P$, where P may be as large as nine in our examples. For $p > 1$, the baseline model is simply modified by changing bounds on variables and changing parameters associated with variables in the objective function, and then resolved. GAMS does not have this capability, and must regenerate DCOPF for each p. Thus, by using Mosel, we avoid regenerating $100\% \times (P-1)/P$ of the model instances that must be solved. Furthermore, Xpress generates an individual model instance much faster than GAMS. For example when using the ERCOT grid (see Chapter IV Section C), GAMS requires approximately ten seconds to generate an instance of DCOPF while Xpress requires only two seconds.

C.   IMPLEMENTATION OF THE IDCH MASTER PROBLEM

The implementation of the IDCH master problem, IAMP, follows the mathematical formulation of Appendix B. Dynamic arrays help create the master problem more efficiently in Mosel than is possible in GAMS. The following lines of code show how to accomplish this for a model that only interdicts buses:

```
declarations
    Delta_bus: dynamic array(Bus) of mpvar;
end-declarations
```

```
forall(i in Bus | B_interdictable(i)) do
      create(Delta_bus(i));
      Delta_bus(i) is_binary;
end-do
```

(In Xpress-MP, when dynamic arrays are used for decision variables, each of them must be "created" explicitly.)

To prevent the master problem from identifying a particular interdiction plan more than once, the following constraints are included at iteration τ of IDCH:

$$\sum_{k|\hat{\delta}_k^t=1} \delta_k \leq \sum_k \hat{\delta}_k^t - 1 \qquad \text{for } t = 2,...,\tau-1 \tag{3.1}$$

The remaining constraints for IAMP are defined in the initial generation of IAMP to find the first interdiction plan. But, all constraints in IAMP from iteration $\tau-1$ to iteration τ are the same, except that one instance of Equation (3.1), for $\tau = \tau-1$, is added. Only this constraint needs to be generated in iteration τ, as shown by the Mosel code below. (Remarks: `Int_Overlap()` is a dynamic array of constraints, corresponding to Equation (3.1), one of which is created in each iteration. `Gen`, `Bus`, `Sub`, and `Line` are sets for the individual components, `All_delta_xxx` is the array where former interdiction plans are stored for components of type xxx, `Delta_xxx` is the interdiction decision variable for component xxx, and `X_interdictable` is a true/false structure. `Ind_iter` corresponds to the iteration counter τ.)

```
declarations
      Int_Overlap: dynamic array(Iter) of linctr;
end-declarations

if(Ind_iter > 2) then
      Int_Overlap(Ind_iter-1) := sum(g in Gen |
      exists(All_delta_gen(Ind_iter-1,g)) and G_interdictable(g))
      Delta_gen(g) + sum(l in Line |
      exists(All_delta_line(Ind_iter-1,l)) and
      L_interdictable(l)) Delta_line(l) + sum(i in Bus |
      exists(All_delta_bus(Ind_iter-1,i)) and
      B_interdictable(i)) Delta_bus(i) + sum(s in Sub |
      exists(All_delta_sub(Ind_iter-1,s)) and
      S_interdictable(s)) Delta_sub(s) <= sum(g in Gen |
      exists(All_delta_gen(Ind_iter-1,g)) and
      G_interdictable(g)) 1 + sum(l in Line |
      exists(All_delta_line(Ind_iter-1,l)) and
      L_interdictable(l)) 1 + sum(i in Bus |
      exists(All_delta_bus(Ind_iter-1,i)) and
      B_interdictable(i)) 1 + sum(s in Sub |
      exists(All_delta_sub(Ind_iter-1,s)) and
```

15

```
        S_interdictable(s)) 1 - 1;
    end-if
```

D.   EFFICIENCY OF XPRESS-MP VERSUS GAMS

The main goal of this thesis is to develop a more effective IDCH by reducing its run time. The last few sections describe why implementing IDCH in Xpress should be more efficient than implementing it in GAMS, provided that solve times are comparable in both cases. To demonstrate this improvement, this section compares run times for both versions of IDCH on a 2.0 GHz Pentium IV computer having 1 GB of RAM. First, we investigate two separate data sets and run the heuristic for 50 iterations while varying the amount of interdiction resource. The datasets for Electric Reliability Council of Texas (ERCOT) and Western Electricity Coordinating Council (WECC) are taken from test cases prepared by the North America Electric Reliability Council as part of a database available from Powerworld [2005]. The run times are presented in Table 1.

Dataset	Total Resources	GAMS/CPLEX Run Time	Xpress Run Time	Reduction in Run Time for Xpress (%)
ERCOT	5	57 min	16 min	71.93%
ERCOT	10	58 min	15 min	74.14%
ERCOT	15	33 min	13 min	60.61%
ERCOT	20	59 min	16 min	72.88%
WECC	5	3 hrs 27 min	40 min	80.68%
WECC	10	3 hrs 28 min	32 min	84.62%
WECC	15	2 hrs 20 min	28 min	80.00%
WECC	20	3 hrs 29 min	42 min	80.82%

Table 1.   Run times for GAMS/CPLEX versus Xpress. These times are for 50 iterations of IDCH applied to the ERCOT and WECC datasets.

The reduction in run times shown in Table 1, between Xpress and GAMS/CPLEX, does not result from differences in solution times for the individual

16

solvers. In fact, CPLEX solves IAMP about five seconds faster than Xpress-MP, on average, while both solve DCOPF in approximately the same amount of time. As a result, Xpress's efficiency in model generation, and the ability to solve a previously generated model without regenerating it, accounts for the reduction in run times exhibited in Table 1.

Table 1 shows that Xpress reduces run times significantly compared to GAMS/CPLEX. The smallest reduction is 60%, the largest is 85%, and the average reduction is 75.7%. The run times in Table 1 for GAMS/CPLEX are calculated using the "solvelink = 1" option. This option allows GAMS to remain open while the solver is running. The GAMS default is to close when the solver is running in order to conserve memory. Run times for GAMS increase by about 30% if the default option is used with these data sets.

While Table 1 shows how Xpress improves run times for IDCH, a more relevant comparison of the Xpress and GAMS/CPLEX implementations may be made by comparing answers to this question: How good is the solution obtained by IDCH in the amount of time an analyst would like to wait for an answer? To make this comparison, we run both versions of the IDCH for 15 minutes and compare the best interdiction plans found in terms of unmet demand. We use the ERCOT data for the comparison; see Table 2.

Total Resources	Xpress: Unmet Demand (MWh)	GAMS: Unmet Demand (MWh)	Xpress: Iterations Completed	GAMS: Iterations Completed
5	40,413	24,167	47	13
10	114,184	90,947	50	13
15	427,924	160,832	58	23
20	415,991	356,996	47	13

Table 2. Unmet demand when IDCH runs for 15 minutes on the ERCOT data, using Xpress or GAMS. (Xpress's solution with 20 units of resource is worse than that given 15 units because IDCH is not guaranteed to find optimal solutions.)

Table 2 above shows that being able to complete more iterations in 15 minutes allows Xpress to find a better interdiction plan than GAMS. On average, in 15 minutes, the Xpress implementation is able to identify a solution that causes 68.8% more disruption (unmet demand) than the GAMS implementation. Note that the unmet demand with 15 units of resource is greater than with 20 units. This simply shows that the IDCH does not always find an optimal solution. When IDCH runs with 20 units of resource, but starts with the best interdiction plan found in 15 minutes given 15 units of resource, it finds a plan that yields 450,595 MWh of unmet demand.

E. XPRESS-MP PROGRAMMING TIPS

The original plan for reimplementing IDCH was to take advantage of Xpress's ability to solve related mathematical-programming problems multiple times with only a single model-generation step. However, if Mosel has generated two models, namely DCOPF and IAMP, and the `sethidden()` command (described below) is not used, the constraints of both models must be satisfied when solving either. This results in long run times. It is better to use Mosel's `sethidden()` command to hide the constraints of the model that is not being solved. Unfortunately, any time this command is used, the model of interest must be regenerated. This drawback is offset by Mosel being more efficient than GAMS at model generation; thus generating the master problem and DCOPF once per iteration does not substantially increase overall run times in Xpress. By taking advantage of solving a modified problem without regenerating the model, the total generation time of the DCOPF is reduced by a factor of $\frac{P-1}{P}$ compared to regenerating each of P DCOPF subproblems in every iteration of the heuristic in GAMS.

Mosel makes efficient use of sparse linked lists, but the programmer must be careful about the related issue discussed here. IDCH must calculate the total flow across a bus. The following calculates "positive flow" for bus i:

```
forall(i in Bus) do
      Positive_Flow(i) := sum(l in Line |
      (exists(Line_O(i,l)) and P_line_sol(l) > 0))
      P_line_sol(l);
end-do
```

where `exists()` enables the user to access a sparse linked list by only looking at the elements in that list. If `exists` were not used, the `sum(l in Line|...)` would be taken over all lines, instead of just the few whose origin is bus `i`. However, what we really want to calculate is the total flow out of bus *i*, which is the sum of positive flows on lines that have bus *i* as an origin plus the absolute values of negative flow on lines that have bus *i* as a destination. That can be written as:

```
forall(i in Bus) do

    Total_flow_out(i) := sum(l in Line |
    (exists(Line_D(i,l)) and P_line_sol(l) < 0) or
    (exists(Line_O(i,l)) and P_line_sol(l) > 0))
    abs(P_line_sol(l));
end-do
```

However, this does not work as one might expect, and, in fact, `sum(l in Line|...)` is taken over all lines. When handling the computation in this manner, it takes IDCH 72 seconds to complete an iteration using the ERCOT data set. This calculation does not utilize the efficiency gained with dynamic arrays because of the `or`. However, the `or` can be equivalently replaced with a sum:

```
forall(i in Bus) do
    Total_flow_out(i) := sum(l in Line |
    exists(Line_D(i,l)) and P_line_sol(l) < 0)
    abs(P_line_sol(l)) + sum(l in Line |
    exists(Line_O(i,l)) and P_line_sol(l) > 0)
    abs(P_line_sol(l));
end-do
```

This allows us to take advantage of the efficiency gained by the `exists()` command and results in the time per iteration dropping to 12 seconds.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. COMPARING IDCH AND RULE-BASED HEURISTICS FOR IDENTIFYING EFFECTIVE INTERDICTION PLANS

This chapter describes the techniques that Albert, Albert and Nakarado [2004] (AAN) use for analyzing the "Structural Vulnerability of the North American Power Grid" and compares their effectiveness to IDCH.

A. GRID FUNCTIONALITY

AAN attempt to analyze the vulnerability of the North American electrical power grid to terrorist attacks by using a surrogate for the functionality of the network rather than direct, electrical-engineering measures, as used in IDCH. They divide the grid into substations (i.e., nodes) and lines (i.e., arcs) to represent a network. The substations are further divided into generating substations (directly connected to sources of power), transmission substations (these transfer the power among high-voltage lines), and distribution substations (at the outer edge of the transmission grid, connected by a single high-voltage line). Our grid data contain additional details on components such as transformers and buses, which allows for a more detailed analysis.

To develop a surrogate for grid functionality, AAN use "an idealized view" of the grid that ignores constraints on line capacities, generator capacities and phase angles. They assume that a load (demand) can be met as long as that load is connected to at least one generating substation via at least one uninterdicted path of lines and substations. To measure the surrogates' effectiveness, AAN uses the concept of connectivity loss, C_L, defined as

$$C_L = 100\% \times \left(1 - \frac{\sum_i N_g^i(\boldsymbol{\delta})}{\sum_i N_g^i(\boldsymbol{0})} \right)$$

where the index i corresponds to distribution substations, $N_g^i(\boldsymbol{\delta})$ denotes the number of generating substations that substation i is connected to given interdiction plan $\boldsymbol{\delta}$, and $N_g^i(\boldsymbol{0})$ is the corresponding value given the null interdiction plan, i.e., under normal

circumstances.   (It turns out that all generators are connected to all distribution substations in the AAN data, so the denominator equals the number of generators times the number of distribution substations.)

Clearly, connectivity loss is not a good measure of grid functionality, or the loss thereof, because it ignores the electrical engineering realities of the grid. We use a more appropriate measure, "fraction of unmet demand" (FUD), defined as

$$\text{FUD} = \frac{Unmet\ Demand\ for\ Energy}{Total\ Demand\ for\ Energy} .$$

FUD must be defined over some "period of study," which nominally corresponds to the longest repair time of any interdicted component.

B.   HEURISTIC RULES FOR IDENTIFYING CRITICAL COMPONENTS IN AN ELECTRICAL GRID

AAN attempt to identify sets of substations whose interdiction would most seriously degrade system functionality. To do this, they sequentially select substations for interdiction using three heuristic rules. A set of distribution substations S_1 is "more critical" than subset S_2 if interdiction of the substations in S_1 leads to a higher value of C_L than does interdiction of the substations in S_2.   We will use their definition of "criticality," except that system functionality will be measured through FUD.

1.   Degree-based Interdiction

The first rule interdicts transmission substations in decreasing order of "substation degree." The degree of a substation is the number of lines that are incident to it. This seems like a sensible rule because power grids are designed to have a certain level of redundancy, and targeting substations with the highest degree would seem to reduce this redundancy quickly. We implement this rule, but measure how well it performs (for the terrorists) in terms of FUD instead of C_L. Using this rule, AAN find that interdicting 5% of the transmission substations in their grid causes a connectivity loss of 45%. This translates to the average transmission substation connecting to 55% of the grid's generators.

22

2. Static Pseudo-load-based Interdiction

The second rule is based on what AAN call "load" at a transmission substation, but which we will call "pseudo-load." ("Load" has a specific meaning in electric power engineering and the use of "pseudo-load" avoids confusion.) They assume that power is "routed" through the most direct path (i.e., from all generation substations to all other reachable substations), and define pseudo-load as the number of shortest paths from a generator to a distribution substation. The "static pseudo-load-based interdiction rule" computes pseudo-loads once at the transmission substations and then interdicts those substations in decreasing order of pseudo-load. This rule is "static" because pseudo-load is not recomputed after each interdiction. Using this rule, AAN find that interdicting 5% of the substations in their grid causes a connectivity loss of 60%. This translates to the average transmission substation connecting to 40% of the grid's generators.

We implement an analogous rule based on the actual load integrated over the LDC at each transmission substation, as calculated by the uninterdicted DCOPF model.

3. Dynamic Pseudo-load-based Interdiction

This rule is the same as the static pseudo-load-based rule above, except that after every ten interdictions, pseudo-load is recomputed. (AAN use the term "cascading" to describe this rule. We avoid using this term because of the confusion it might cause with the normal use of "cascading outages" in the power-engineering literature.) AAN recalculate pseudo-load after every ten interdictions, rather than after each interdiction, because of computational expense. Following this rule, AAN find that interdicting 5% of the transmission substations in their grid causes a connectivity loss of 90%. This translates to the average transmission substation connecting to only 10% of the grid's generators.

We implement an analogous rule using actual loads integrated over the LDC, but recalculate these loads, using DCOPF, after the interdiction of each subsequent substation.

4. Extent of Possible Attacks

The substations in our electrical grids are all transmission substations, so this thesis allows any substation to be interdicted. The substations are not part of the original NERC data, but VEGA identifies them by grouping adjacent transformers that are

connected to buses whose nominal voltages are 69kV and above. We assume no special security measures have been instituted at any substation because (a) we have no information about such measures, and (b) we want to compare to AAN, who assume no substations are protected.

AAN evaluate connectivity loss in their grid data with up to 10% of the substations being interdicted. But, in their data this translates into as many as 1,028 substations being attacked, which is clearly unreasonable. We assume that the best organized and most well-funded terrorist group could attack about 25 substations. (Even this seems unlikely, but it should bound the worst-possible situation.)

C.   ANALYSIS OF THE ERCOT ELECTRICAL GRID

The first comparison of AAN's techniques to IDCH uses the ERCOT electrical grid. The ERCOT data cover 4,993 lines, 946 transformers, 4,923 buses, 474 generators, and 499 substations. We also allow the interdiction of buses that lie outside of substations. However, the largest degree of such a bus is 10 and there are 26 substations with degree of 11 or greater. Therefore, this point is moot for the degree-based rule, and only substations will be interdicted.

The degree-based rule is implemented to break ties arbitrarily, and there are ties in the ERCOT data. As a result, FUD is not unique when computed by this rule except for the last time a substation of a given degree is interdicted. In this scenario, unique values result from interdicting 1-5, 7, 11, 19, or 26 substations. Figure 1 shows the value of FUD for ERCOT using the three rules proposed by AAN and by using IDCH.
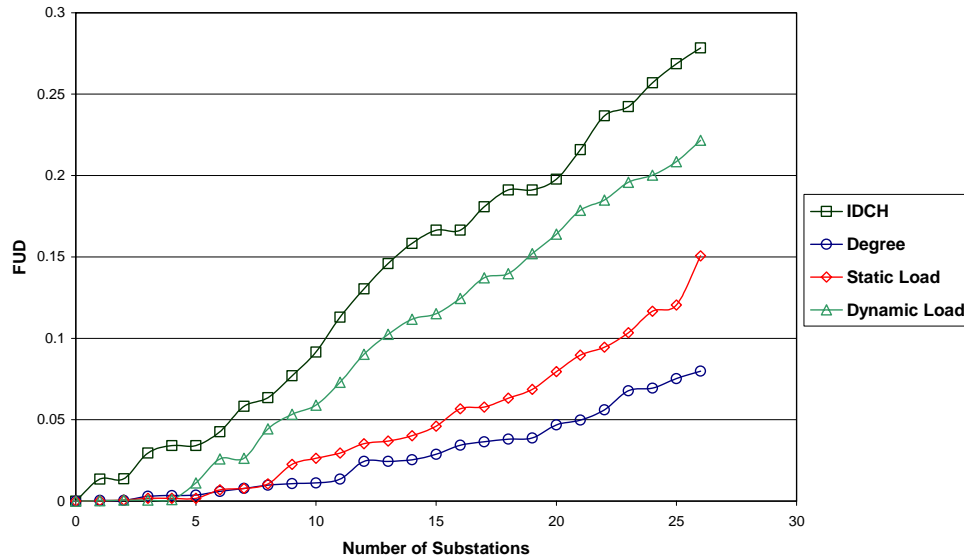
Figure 1.   Fraction of Unmet Demand, FUD, in the ERCOT grid while interdicting substations using rules adapted from Albert, Albert and Nakarado [2004] and using IDCH. IDCH finds the most disruptive set of substations to interdict in all cases.

Figure 1 shows that IDCH identifies the most disruptive interdiction plan for the ERCOT grid. That is, the rules suggested by AAN lead to serious underestimates of the disruption that a well-funded, sophisticated terrorist group might cause. And, in other words, the grid is more vulnerable to attack than AAN's method would lead an SO to believe. Less sophisticated terrorists would want to use the dynamic load-based rule to guide their interdictions, and even less-sophisticated ones would want to use the static load-based rule. The worst rule, for the terrorists, is the degree-based rule.

The best rule presented by AAN requires the interdiction of at least five substations before the terrorists achieve a FUD greater than 1%. IDCH demonstrates a significant disruption by interdicting just a single substation. If an SO were to use this type of analysis to decide which substations would benefit most from having extra security, IDCH would clearly provide the better tool.

D. ANALYSIS OF THE WECC ELECTRICAL GRID

We apply the same techniques as in the ERCOT example to the WECC grid. The WECC data cover 7,872 lines, 3,069 transformers, 8,436 buses, 1,400 generators, and 1,264 substations. The WECC data provided by NERC, exhibits more component aggregation ("equivalencing") than the ERCOT data does. This explains why the WECC data size is only twice the size of ERCOT data, despite the fact that WECC covers a geographical area at least five times that of ERCOT.

Once again, we allow the interdiction of buses that lie outside of substations. The largest degree of such a bus is 13 and there are 24 substations with degree of 15 or greater. We limit the total number of interdictions to 24; therefore, no buses are interdicted with the degree-based rule. FUD is unique when evaluating 2-5, 9, 12, 17, or 24 substations with this rule. Figure 2 displays results.

For simplicity, this analysis ignores extra constraints in WECC referred to as "nomogram limits." More realistic studies should include these constraints because their omission could lead to optimistic results for the SO. That is, the disruption levels identified here might be somewhat worse if the nomogram limits are enforced. See Appendix C for a discussion of nomogram limits.
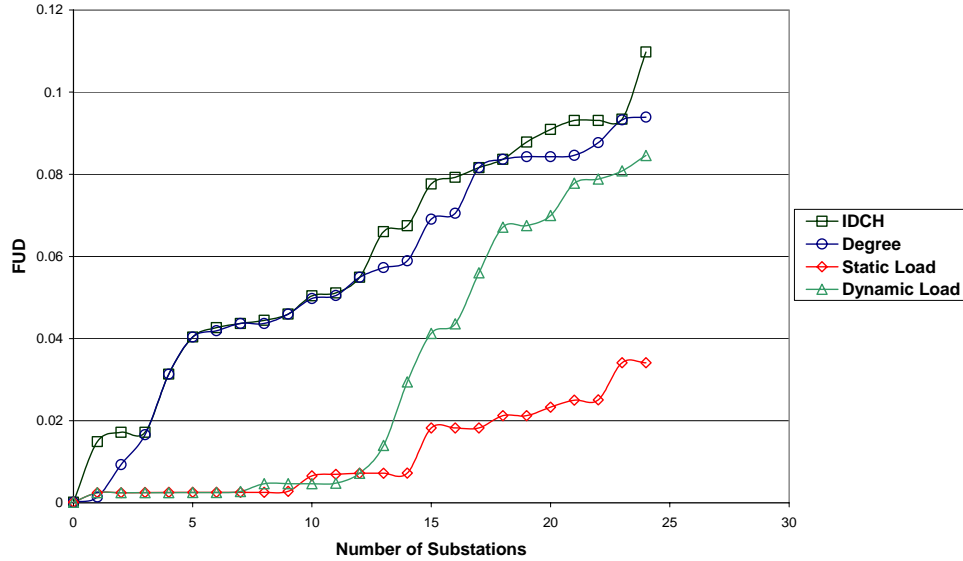
Figure 2.   Fraction of Unmet Demand, FUD , in the WECC grid while interdicting substations using rules adapted from Albert, Albert and Nakarado [2004] and using IDCH. Unlike ERCOT, the degree-based rule works well here, and is almost as good as IDCH.

Figure 2 shows that the load-based rules are worst for WECC. In contrast to ERCOT, the degree-based algorithm works well, nearly as well as IDCH. Our WECC data cover the peak demand period of the LDC during summer months. Consequently, the grid is operating near maximum capacity. We believe the high demand reduces the grid's effective redundancy and explains why the degree-based rule works so well here. AAN find that their dynamic pseudo-load algorithm is the best, followed by the pseudo-load algorithm and lastly, the degree-based algorithm. Our adaptations of these strategies in the WECC grid show that such general conclusions do not apply when the physical realities of power flows are considered.

E.   INTERDICTING LINES IN AN ELECTRICAL GRID

AAN only investigate the effects of interdicting substations. While removing substations from a grid would be very damaging, substations are relatively easy to protect from attack, at least in theory, because of their small "footprint." For similar reasons,

buses and generators are easy to protect. Transmission lines, on the other hand, run for hundreds of thousands of miles and are therefore difficult to protect.

Although lines are easy to repair, because of their vulnerability, we believe that it is important to investigate their possible interdiction. Attacking lines is, in fact, simple, as demonstrated by Colombian FARC terrorists, who have destroyed thousands of 230-kV and 500-kV towers since 1985. ISA spokesman acknowledged, "We can't post a soldier at every tower" [Miami Herald 2002]. The last of these attacks in September 2005, destroyed six towers in the Cauca area, affecting over three million people for several days and requiring emergency restoration plans [El Pais-Cali Colombia 2005]. We investigate several heuristic rules for identifying the most critical sets of lines, and also apply IDCH. The three heuristic rules are:

1. Capacity-based interdiction: Intuitively, a line's capacity seems like it might be a good indicator of its criticality. Since it is easy to distinguish between high-capacity and low-capacity lines, a group of terrorists could plausibly use this strategy for attacking a grid: Interdict the highest-capacity lines first, breaking ties arbitrarily.

2. Static load-based interdiction: This rule is identical to "static load-based interdiction" as applied to substations, except that load now refers to the "load" (power averaged across the LDC) being carried by lines. The static load-based rule interdicts lines in decreasing order of the average power they carry as calculated by the uninterdicted DCOPF.

3. Dynamic load-based interdiction: This rule is the same as (2) above, except that "load" is recomputed after each interdiction.

As IDCH does, we treat lines that are physically parallel (i.e., mounted on the same tower) as single lines, subject to a single interdiction. And, of course, the load on the "composite line" is the sum of the individual loads. We examine the effects of line interdiction on FUD for the ERCOT dataset and assume that at most 25 lines would be interdicted. Figure 3 displays results.
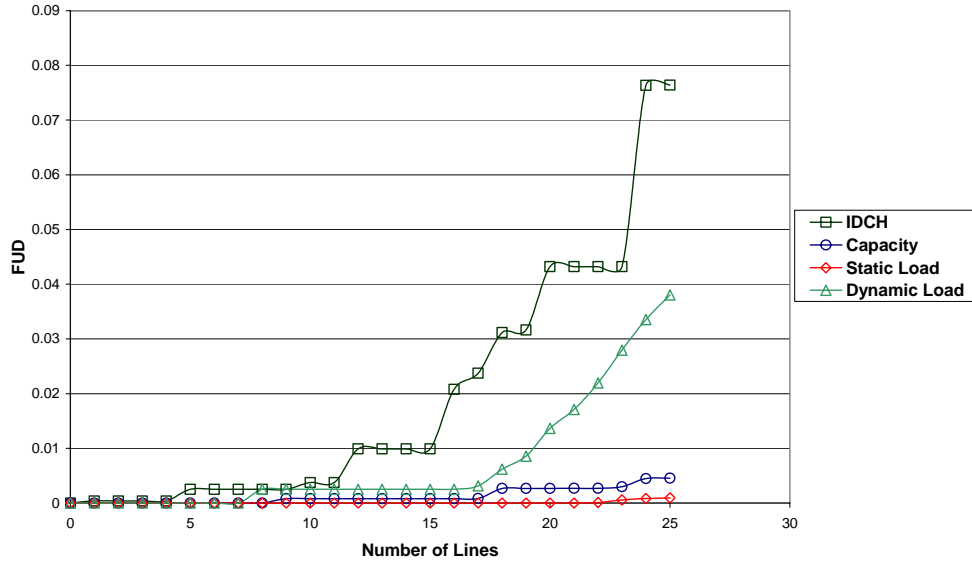
Figure 3.   Fraction of Unmet Demand, FUD, in the ERCOT network while interdicting lines using rule-base algorithms and IDCH. IDCH finds the most disruptive set of lines to interdict in all cases.

Figure 3 shows that the static load-based rule is the worst rule for interdicting lines, and that the capacity-based rule is not much better. The dynamic load-based rule does start to show significant effects as the number of interdicted lines increases. IDCH identifies the most disruptive interdiction plan for the grid, and results obtained using the AAN-based rules underestimate the grid's vulnerability to attack. IDCH shows that the ERCOT grid is somewhat vulnerable to line interdiction: Interdicting only 25 of 4,993 lines, about 0.5%, results in an average of 9,953 MW being shed over 48 hours. This shedding exceeds 7% of the grid's average demand.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

The VEGA software package is an integrated system for analyzing the vulnerability of electric power-transmission grids to terrorist attacks. At VEGA's core is an interdiction model that posits terrorists with limited offensive resources. The goal of the model is to find near-optimal attacks by using the information provided by power flow models for multiple grid configurations under different load levels. This thesis reduces solution times for the Interdicting DC Optimal Power-Flow Heuristic (IDCH) by implementing the algorithm using Xpress-MP, a powerful optimization software package. Additionally, the thesis compares the effects of IDCH's interdiction plans to the effects of interdiction plans created using simple, heuristic rules, as suggested in the literature.

Our new implementation of IDCH in Xpress-MP reduces solution times by 75.7% on average compared to the previous implementation in GAMS when GAMS uses the highly efficient CPLEX solver. This now means that IDCH will often run in minutes rather than hours. We also find that the new implementation provides better solutions when run for a fixed length of time, because it can complete more iterations. For instance, in 15 minutes of run time, the Xpress-MP implementation identifies an interdiction plan that causes 68.8% more disruption (unmet demand for energy) than does the GAMS-identified plan in the same amount of time: Xpress-MP completes 47 iterations in this time while GAMS completes only 13 iterations.

In addition to basic testing the new IDCH implementation has been exercised by analyzing the vulnerability of two real-world electrical power grids to terrorist attacks on its substations. The grid data are provided by the North America Electric Reliability Council and cover the ERCOT grid (Electric Reliability Council of Texas) and the WECC grid (Western Electricity Coordinating Council). IDCH's effectiveness is evaluated by comparing the interdiction plans it produces to those produced using adaptations of the heuristic rules proposed by Albert et al. [2004]. These rules include: (a) interdict the substation with the highest degree (most connected) first, (b) interdict the substation carrying the greatest baseline load first, and (c) interdict the substation carrying the greatest load first, but recompute that load after every interdiction. We find

that IDCH performs substantially better than the heuristic rules, although the degree-based rule works surprisingly well for the WECC grid. In contrast, it is the worst rule for ERCOT. Thus, IDCH produces much more consistent results than the rules from Albert et al.

We also investigate how interdicting lines affects the ERCOT grid. IDCH finds that by interdicting as few as 25 lines results in 7.5% of the demand for energy going unmet. All of the rules suggested by Albert et al. perform poorly, in comparison.

LIST OF REFERENCES

Albert, R., Albert, I., and Nakarado, G. (2004). "Structural Vulnerability of the North American Power Grid," *Physical Review E*, 69, 025103(R), pp. 1-4.

Bard, J.F. and Moore, J.T. (1990). "The Mixed Integer Linear Bilevel Programming Problem," *Operations Research 38*, pp. 911-921.

Dash (2005). Xpress-MP Overview, http://www.dashoptimization.com/home/products/products_overview.html (accessed September 2005)

El Pais-Cali Colombia (2005). "Adoptan plan de emergencia para restablecer energía," http://elpais-cali.terra.com.co/paisonline/notas/Septiembre062005/A4061.html (accessed September 2005)

Energy Pulse (2003). "Power-Grid Independence Means Better Homeland Security," http://www.energypulse.net/centers/article/article_display.cfm?a_id=152 (accessed June 2005).

GAMS (2001). "Secure Work Files," http://www.gams.com/docs/privacy.pdf (accessed September 2005)

GAMS (2004). Software available from www.gams.com (accessed August 2005).

GAMS-CPLEX (2004). http://www.gams.com/solvers/solvers.htm#CPLEX (accessed August 2005).

Heipcke, Susanne (2002). *Applications of Optimization with Xpress^{MP}*, Dash Optimization Ltd., United Kingdom. http://www.dashoptimization.com/home/downloads/book/booka4.pdf (accessed September 2005)

Miami Herald (2002). ISA spokesman, 6 March 2002.

NERC (North American Electric Reliability Council) (2002). "The Electricity Sector Response to the Critical Infrastructure Protection Challenge," ftp://www.nerc.com/pub/sys/all_updl/cip/ES-reposnse-to-CIP.pdf (accessed September 2005)

North American Energy Working Group (2002). "North America The Energy Picture," http://www.eia.doe.gov/emeu/northamerica/enginfr3.htm (accessed September 2005).

Powerworld (2005). One-Line Diagrams and FERC 715 Cases, http://www.powerworld.com/ferc/ferc715.asp (accessed September 2005)

Report of the National Energy Policy Development Group (2001). "Reliable, Affordable, and Environmentally Sound Energy for America's Future," http://www..whitehouse.gov/energy/National-Energy-Policy.pdf (accessed September 2005).

Salmeron, J., Wood, K. and Baldick, R. (2004A). "Optimizing Electric Grid Design Under Asymmetric Threat (II)," Technical Report NPS-OR-04-001, Naval Postgraduate School, Monterey, California.

Salmeron, J., Wood, K. and Baldick, R. (2004B). "Analysis of Electric Grid Security Under Terrorist Threat," *IEEE Transactions on Power Systems*, Vol. 19, pp. 905-912.

Salmeron, J., Wood, K. and Baldick, R. (2005). "Optimizing Electric Grid Design Under Asymmetric Threat (III)," Technical Report NPS-OR-05-005, Naval Postgraduate School, Monterey, California. (Restricted Distribution)

Xpress-Mosel Language Reference Manual Release 1.2 (2003). http:// computing.ee.ethz.ch/sepp/xpress-13b-et/images/manuals.html (accessed December 2004).

Xpress-Mosel User Guide (2004). http:// computing.ee.ethz.ch/sepp/xpress-13b-et/images/manuals.html (accessed December 2004).

Xpress-MP Essentials. http:// computing.ee.ethz.ch/sepp/xpress-13b-et/images/manuals.html (accessed December 2004).

Xpress-MP Getting Started (2003), Dash Optimization Ltd., United Kingdom.

APPENDIX A.   MATHEMATICAL FORMULATION OF DCOPF

The single period DC Optimal Power Flow Model from Salmeron et al. [2005] follows:

Sets:

$i \in I$ electrical buses

$i \in I^0$ reference buses, $I^0 \subseteq I$

$g \in G$ generating units

$g \in G_i$ generating units connected to bus i, $G_i \subseteq G$

$l \in L$ transmission lines and transformers

$l \in L^{AC}$ AC transmission lines and transformers modeled as AC lines, $L^{AC} \subseteq L$

$l \in L^{DC}$ DC transmission lines, $L^{DC} \subseteq L$

$l \in L_i^{Bus}$ lines connected to bus i (AC and DC), $L_i^{Bus} \subseteq L$

$l \in L_l^{Par}$ lines in parallel with line l, $L_l^{Par} \subseteq L$

$c \in C$ consumer sectors (e.g., residential, industrial)

$s \in S$ substations

$i \in I_s$ buses at substation s, $I_s \subseteq I$

$l \in L_s$ lines connected to substation s (including transformers, which are represented by lines), $L_s \subseteq L$

$g \in G_s$ generators at substation s, $G_s \subseteq G$

Parameters and [units] if applicable:

$o(l), d(l)$ origin and destination buses, respectively for line l (more

         than one line with the same $o(l)$, $d(l)$ may exist)

$i(g)$ unique bus connected to generator g, i.e., $g \in G_{i(g)}$

$s(i)$ substation $s \in S$ that contains bus $i \in I_s$ (not all buses are

         contained in substations)

d_{ic} load of consumer sector c at bus i [MW]

$\overline{P}_l^{Line}$ transmission capacity for AC or DC line l [MW]

$\overline{P}_g^{Gen}$ maximum output from generator g [MW]

$\underline{P}_g^{Gen}$ minimum output from generator g [MW]

r_l, x_l resistance and reactance of AC line l, respectively [p.u.] (We

         assume $x_l \gg r_l$)

B_l series susceptance for AC line l, calculated as $B_l = x_l / \left(r_l^2 + x_l^2 \right)$

         [p.u.]

R_l, P_l, E_l resistance $[\Omega]$, set point [MW] and voltage [kV] for DC line l,

         respectively

μ_l transmission coefficient (= 1 – loss coefficient) on DC line l,

         calculated as $\mu_l = 1 - I^2 R / P = 1 - P^2 R / \left(E^2 P \right) = 1 - PR / E^2$ [p.u.]

h_g generation cost for generator g [$/MWh]

f_{ic} load-shedding cost for customer sector c at bus i [$/MWh]

Decision variables [units]:

P_g^{Gen} generation from unit g [MW]

P_l^{Line} power flow on AC line l [MW]

U_l power flow from the "from" to the "to" bus for DC line l [MW].

(DC lines are modeled as follows: If $U_l \geq 0$ MW are sent from the

"from" bus, then $(1-\mu_l)U_l$ MW are received at the "to" bus.)

V_l power flow from the "to" to the "from" bus of DC line l [MW].

(Similarly, if $V_l \geq 0$ MW are sent from the "to" bus, then $(1-\mu_l)V_l$

MW are received at the "from" bus.)

S_{ic} load shed (unmet demand) for customer sector c at bus i [MW]

θ_i phase angle at bus i [radians]

Formulation:

DCOPF: $\displaystyle\min_{P^{Gen},P^{Line},S,\theta,U,V} \sum_g h_g P_g^{Gen} + \sum_i \sum_{c|d_{tic}>0} f_{ic} S_{ic}$ \hfill (DCOPF.1)

s.t.

$$P_l^{Line} = B_l\left(\theta_{o(l)} - \theta_{d(l)}\right) \qquad\qquad \forall l \in L^{AC} \qquad\qquad \text{(DCOPF.2)}$$

$$\sum_g P_g^{Gen} - \sum_{\substack{l\in L^{AC}|\\o(l)=i}} P_l^{Line} + \sum_{\substack{l\in L^{AC}|\\d(l)=i}} P_l^{Line} + \sum_{\substack{l\in L^{DC}|\\o(l)=i}} \left(-U_l + \mu_l V_l\right) + \sum_{\substack{l\in L^{DC}|\\d(l)=i}} \left(\mu_l U_l - V_l\right) =$$

$$\sum_{c|d_{tic}>0} \left(d_{ic} - S_{ic}\right) \qquad\qquad \forall i \in I \qquad\qquad \text{(DCOPF.3)}$$

$$-\overline{P}_l^{Line} \leq P_l^{Line} \leq \overline{P}_l^{Line} \qquad\qquad \forall l \in L \qquad\qquad \text{(DCOPF.4)}$$

$$\underline{P}_g^{Gen} \leq P_g^{Gen} \leq \overline{P}_g^{Gen} \qquad\qquad \forall i, \forall g \in G_i \qquad\qquad \text{(DCOPF.5)}$$

$$0 \leq S_{ic} \leq d_{ic} \qquad\qquad \forall i,c \mid d_{ic} > 0 \qquad\qquad \text{(DCOPF.6)}$$

$$\theta_i = 0 \qquad\qquad \forall i \in I^0 \qquad\qquad \text{(DCOPF.7)}$$

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B.   MATHEMATICAL FORMULATION OF IAMP (INTERDICTION-APPROXIMATING MASTER PROBLEM)

The "interdicting approximating master problem" from Salmeron et al. [2005] follows:

Sets:

$G^* \subseteq G$, $L^* \subseteq L$, $I^* \subseteq I$, $S^* \subseteq S$, subsets of interdictable generators, lines, buses, and substations, respectively.

Parameters:

$V_g^{Gen,\tau}$	value for each generator g in iteration τ
$V_l^{Line,\tau}$	value for each line l in iteration τ
$V_i^{Bus,\tau}$	value for each bus i in iteration τ
$V_s^{Sub,\tau}$	value for each substation s in iteration τ
M_g^{Gen}	resources required to interdict generator g
M_l^{Line}	resources required to interdict line l
M_i^{Bus}	resources required to interdict bus i
M_s^{Sub}	resources required to interdict substation s
M	total resources available for interdiction

Decision Variables:

$\delta_g^{Gen,\tau}$	determines if generator g is interdicted in iteration τ
$\delta_l^{Line,\tau}$	determines if line l is interdicted in iteration τ
$\delta_i^{Bus,\tau}$	determines if bus i is interdicted in iteration τ
$\delta_s^{Sub,\tau}$	determines if substation s is interdicted in iteration τ

Formulation:

$$\text{MP}\left(V_k^{\tau}\right): \quad \max_{\delta} V_k^{\tau}\delta_k^{\tau} \tag{MP.1}$$

s.t.:

$$\sum_{g\in G^*} M_g^{Gen}\delta_g^{Gen,\tau} + \sum_{l\in L^*} M_l^{Line}\delta_l^{Line,\tau} + \sum_{i\in I^*} M_i^{Bus}\delta_i^{Bus,\tau} + \sum_{s\in S^*} M_s^{Sub}\delta_s^{Sub,\tau} \leq M \tag{MP.2}$$

$$
\begin{aligned}
\delta_g^{Gen,\tau} + \delta_i^{Bus,\tau} &\leq 1 & \forall g\in G_i\cap G^*, \forall i\in I^* \\
\delta_l^{Line,\tau} + \delta_i^{Bus,\tau} &\leq 1 & \forall l\in L_i\cap L^*, \forall i\in I^* \\
\delta_l^{Line,\tau} + \delta_{l'}^{Line,\tau} &\leq 1 & \forall l'\in L_l^{Par}\cap L^*, \forall l\in L^* \\
\delta_i^{Bus,\tau} + \delta_s^{Sub,\tau} &\leq 1 & \forall i\in I_s\cap I^*, \forall s\in S^* \\
\delta_l^{Line,\tau} + \delta_s^{Sub,\tau} &\leq 1 & \forall l\in L_s\cap L^*, \forall s\in S^* \\
\delta_g^{Gen,\tau} + \delta_s^{Sub,\tau} &\leq 1 & \forall g\in G_s\cap G^*, \forall s\in S^*
\end{aligned}
\tag{MP.3}
$$

$$
\begin{aligned}
&\sum_{\substack{g\in G^*| \\ \delta_g^{Gen,\tau'}=1}} \left(\delta_g^{Gen,\tau'}-\delta_g^{Gen,\tau}\right) + \sum_{\substack{l\in L^*| \\ \delta_l^{Line,\tau'}=1}} \left(\delta_l^{Line,\tau'}-\delta_l^{Line,\tau}\right) + \\
&\sum_{\substack{i\in I^*| \\ \delta_i^{Bus,\tau'}=1}} \left(\delta_i^{Bus,\tau'}-\delta_i^{Bus,\tau}\right) + \sum_{\substack{s\in S^*| \\ \delta_s^{Sub,\tau'}=1}} \left(\delta_s^{Sub,\tau'}-\delta_s^{Sub,\tau}\right) \geq 1
\end{aligned}
\qquad \forall \tau'=1,\dots,\tau-1 \tag{MP.4}
$$

$$\delta_g^{Gen,\tau}, \delta_l^{Line,\tau}, \delta_i^{Bus,\tau}, \delta_s^{Sub,\tau} \in \{0,1\} \qquad \begin{aligned}&\forall g\in G^*, \forall l\in L^*, \\ &\forall i\in I^*, \forall s\in S^*\end{aligned} \tag{MP.5}$$

At each iteration τ, (MP.1) interdicts the most valuable components based on the V_k^{τ} for each component k. (MP.2) prevents interdicting more components than total available resources, *M,* allow. (MP.3) prevents interdicting a component that is indirectly interdicted due to being connected to another directly interdicted component. (MP.3) assumes a bus is superior (i.e. takes longer to repair) than a line or generator and a substation is superior to a bus. If an inferior element is in fact superior (MP.3) relaxes for those two elements. (MP.4) prevents selecting an interdiction plan, δ, that has been selected in a previous iteration. (MP.5) forces all decision variables to be binary.

APPENDIX C.   NOMOGRAMS

A nomogram is simply a constraint on two or more lines that reduces the combined capacity to less than the nominal sum of capacities. For two lines, this can be represented by the following constraints:

$$P_{l1}^{Line} \leq \overline{P}_{l1}^{Line}$$
$$P_{l2}^{Line} \leq \overline{P}_{l2}^{Line}$$
$$P_{l1}^{Line} + P_{l2}^{Line} \leq NC_{l1,l2},$$

where $NC_{l1,l2}$ is the combined capacity and $\overline{P}_{l1}^{Line} + \overline{P}_{l2}^{Line} > NC_{l1,l2}$

(Here we assume that positive values of P_{l1}^{Line} and P_{l2}^{Line} reflect flow in the direction where the nomogram constraint applies.) Adding such a nomogram to the DCOPF model is accomplished by the following code:

```
declarations
      Nomo_flow: array(Line,Line) of real;
      Nomo_const: array(Line,Line) of linctr;
end-declarations


forall(l in Line, ll in Line | exists(Nomo_flow(l,ll))) do
      Nomo_const(l,ll) := P_line(l) + P_line(ll) <=
      Nomo_flow(l,ll)
end-do
```

Here, `Nomo_flow(Line,Line)` contains the maximum flow that is allowed across the two lines simultaneously, and we may assume it is read from a data file as the rest of the problem data.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1.   Defense Technical Information Center
        Ft. Belvoir, Virginia

2.   Dudley Knox Library
        Naval Postgraduate School
        Monterey, California

3.   Professor R. Kevin Wood, Code OR/Wd
        Department of Operations Research
        Naval Postgraduate School
        Monterey, California

4.   Professor Javier Salmeron, Code OR/Sa
        Department of Operations Research
        Naval Postgraduate School
        Monterey, California

5.   Professor Ross Baldick
        Department of Electrical Engineering
        University of Texas at Austin
        Austin, Texas

6.   LCDR David D. Carnal
        Commander, U.S. Naval Forces Central Command
        Manama, Bahrain